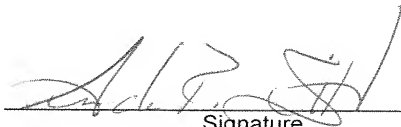


Doc Code: AP.PRE.REQ

PTO/SB/33 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW		Docket Number (Optional) 31146/MP1502	
	Application Number 10/802,586-Conf. #5149	Filed March 17, 2004	
	First Named Inventor Joel D. Munter et al.		
	Art Unit 2192	Examiner B. C. Wang	
<p>Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.</p> <p>This request is being filed with a notice of appeal.</p> <p>The review is requested for the reasons stated on the attached sheets. Note: No more than five (5) pages may be provided.</p> <p>I am the</p> <p><input type="checkbox"/> applicant /inventor.</p> <p><input type="checkbox"/> assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)</p> <p><input checked="" type="checkbox"/> attorney or agent of record. Registration number 62,162</p> <p><input type="checkbox"/> attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34. _____</p> <p> Signature Andrew R. Smith Typed or printed name (312) 474-6300 Telephone number February 22, 2010 Date</p> <p>NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.</p> <p><input type="checkbox"/> *Total of 1 forms are submitted.</p>			

STATEMENT OF SUPPORT OF PRE-APPEAL BRIEF REQUEST FOR REVIEW

The examiner issued a new rejection of claims 1-4, 7-11, 13-18, 20-26, and 28-33 under 35 U.S.C. §103(a) as allegedly unpatentable over Chauvel (U.S. Pub. No. 20040010785) in view of Chheda (US Pub. No. 20050114850). The Examiner's rejection should not be upheld at least for reasons best summarized in a discussion of independent claim 1 below.

Independent claim 1 recites, *inter alia*:

determining an initial number of times to interpretively execute the plurality of non-native instructions;

interpretively executing the plurality of non-native instructions the initial number of times;

monitoring execution of the plurality of non-native instructions to determine when the plurality of non-native instructions have been interpretively executed the initial number of times;

compiling the plurality of non-native instructions to generate object code for the plurality of non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times (emphasis added).

Chauvel describes profiling or estimating performance characteristics of an application during execution on a computing device by comparing an application profile against a virtual machine profile. Chauvel further describes generating a byte-code based profile of the application that indicates how many times the application calls for execution of an operation or method (Chauvel, paragraphs 0012-0013). An API implementing a modified interpreter loop and other modifications to a Java Virtual Machine (JVM) generates the virtual machine profile. The virtual machine profile provides information indicative of a performance characteristic (i.e., execution time, energy consumption, or power level) of the device's underlying hardware for each particular application operation. The information in the application profile and the virtual machine profile are then combined to generate a performance estimate to aid a programmer in optimizing the application or to aid in efficient operation of the device. Chauvel also generally describes that a just-in-time (JIT) compiler may also be employed to produce the application and virtual machine profiles. The generated performance estimate is generally described as used for scheduling operations. Therefore, by comparing an application profile that is based on a number of times an operation appears in a

code segment and a virtual machine profile, Chauvel generally describes generating an application performance estimate so that particular application operations may be re-scheduled before execution to optimize the performance of the device.

The teachings of Chauvel, either alone or in combination with Chheda, do not teach or suggest the recitations of claim 1 discussed above and as alleged in the November 24, 2009 Office Action. In an interview with examiner Ben Wang on December 9, 2008, the parties discussed the rejection of claims 1, 11, 15, 18, 22, and 26 and, in particular, the alleged teachings of Chauvel. At the conclusion of this interview, the examiner agreed that Chauvel did not teach or suggest the recitations of the pending claims.¹ In particular, the examiner agreed that Chauvel generally teaches determining how many times a code segment is executed in a given code segment rather than determining how many times to interpretively execute non-native instructions an initial number of times as generally recited in claims 11, 18, and 26 (and as added by subsequent amendment to claims 1, 15, and 22). In the office action immediately following this interview, the examiner did not allege that the teachings of Chauvel taught or suggested any of claims' 1, 11, 15, 18, 22, and 26 recitations, instead relying solely on the alleged teachings of Chheda and Chen (Energy-Aware Compilation and Execution in Java-Enabled Mobile Devices, 2003 IEEE, pp. 1-8).²

Despite the agreements reached by the applicants' representative and Examiner Wang regarding the alleged teachings of Chauvel, the current Office Action again cites Chauvel as allegedly teaching or suggesting the "determining..." recitation of claim 1. The applicants submit that, as emphasized by the previous interview agreements, the present Office Action continues to misconstrue the teachings of Chauvel with respect to the "determining..." recitation of claim 1. Instead, as discussed above, Chauvel generally describes creating a runtime profile for the execution of compiled code that includes, among other things, relating power consumption to the number of times an individual operation appears within the code. As described on page 15 of the October 15, 2008 Office Action and on page 4 of the present Office Action (applied to the "determining..." claim recitations), Chauvel describes:

an application profile that specifies a number of executions of a plurality of operations used in the specified portion of the

¹ On page 10 of the applicants' response dated December 18, 2008, the applicants provide a summary of the interview with Examiner Wang. No objections to this summary are evident in the record.

² See Office Action dated April 15, 2009, page 2.

application (Chauvel, paragraph [0012]) [and] the application profile based on the number of times operations are executed in the application (Chauvel, paragraph [0013]) (emphasis added).

Rather than merely counting the number of times an operation is repeated (i.e., appears) within a set of instructions, as generally disclosed by Chauvel, claim 1 generally recites determining how many times to execute, interpretively, non-native instructions an initial number of times. An application profile based on counting how many times an operation appears within an application as generally disclosed by Chauvel does not teach or suggest determining an initial number of times to interpretively execute non-native instructions as generally recited in claim 1.

Additionally, Chauvel does not disclose or suggest “monitoring execution of the plurality of non-native instructions to determine when the plurality of non-native instructions have been interpretively executed the initial number of times” as recited in claim 1. The Office Action cites paragraph 0013 of Chauvel as allegedly disclosing this element. But paragraph 0013 merely states that an application profile is generated “based on the number of times operations are executed in the application.” As explained elsewhere in Chauvel, this statement merely refers to determining how many times an operation is performed during execution of the application. Determining how many times an operation is performed is not the same as determining when instructions have been interpretively executed an already determined initial number of times as generally recited in claim 1.

The Office Action also cited paragraph 0031 of Chauvel as allegedly disclosing the “monitoring” element of claim 1. But paragraph 0031 generally refers to determining how much time it takes to execute instructions. *See also Chauvel* at par. 0030 (“[I]t is important to provide enough information to deduce real execution time.”). Determining how much time it takes to execute instructions is not the same as determining when instructions have been interpretively executed a determined initial number of times as generally recited in claim 1.

On page 4, the Office Action admits that Chauvel does not explicitly disclose the “compiling...” element of claim 1. The Office Action then cites the Abstract of Chauvel as disclosing a profiling system to indicate the number of operation executions in an application and a virtual machine profile that indicates the time/energy consumed by the operation on a particular platform. The Office Action offers no explicit reasoning for citing this portion of Chauvel with reference to the “compiling the plurality of non-native instructions to generate

object code for the plurality of non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times” element of claim 1.

Nevertheless, like the portions of Chauvel discussed above, this cited portion of Chauvel merely describes determining how many times an operation is performed when an application is executed and does not disclose or suggest the “compiling the plurality of non-native instructions to generate object code for the plurality of non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times” element of claim 1. Thus, Chauvel does not teach or suggest the recitations of claim 1 and the Office Action’s continued misunderstanding of Chauvel cannot be a basis for the current rejections of claim 1.

Chheda also does not disclose or suggest the above-discussed elements and, in particular, does not disclose or suggest “compiling the plurality of non-native instructions to generate object code for the plurality of non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times” as alleged on pages 5 and 6 of the Office Action. Rather, Chheda generally describes an executable level re-compilation approach. *See Chheda* at par. 0070. Chheda also generally describes redundant micro-operations that may be replaced with related information extracted in the compiler or in the program information to improve energy efficiency. *See Chheda* at par. 0044-0045. While Chheda may generally describe replacing redundant instructions to improve energy efficiency, Chheda discloses nothing regarding interpretively executing non-native instructions, interpretively executing non-native instructions a determined initial number of times, or compiling non-native instructions only after interpretively executing the non-native instructions the determined initial number of times.

In sum, the Examiner has failed to establish a prima facie case of obviousness of claim 1, over Chauvel in view of Chheda at least because: 1) agreements reached during December 9, 2008 interview with Examiner Wang, as summarized in the applicants’ response dated December 18, 2008, indicate that the portions of Chauvel used in the current rejections have been and continue to be misconstrued; 2) instead of “determining an initial number of times to interpretively execute the plurality of non-native instructions,” Chauvel merely describes determining how many times a code segment appears or is executed within a given set of instructions; 3) neither Chauvel nor Chheda discloses or suggests “monitoring

execution of the plurality of non-native instructions to determine when the plurality of non-native instructions have been interpretively executed the initial number of times”; and 4) neither Chauvel nor Chheda discloses or suggests “compiling the plurality of non-native instructions to generate object code for the plurality of non-native instructions to only after interpretively executing the plurality of non-native instructions the initial number of times”. Thus, Chauvel in view of Chheda clearly does not teach or suggest the recitations of claim 1.

At least for reasons similar to those discussed above with respect to claim 1, the Examiner has failed to establish a prima facie case of obviousness of independent claims 11, 15, 18, 22, and 26. With regard to the dependent claims, applicants respectfully submit that the Examiner failed to establish a prima facie case of obviousness of each dependent claim at least for the same reasons as the corresponding independent claims from which the claim depends.